

FR2826753

Publication Title:

Processing of an Extended Markup Language XML document in an Internet network, in which user selects a markup language and a list of its functions is displayed and XML document is associated with Schema XML language

Abstract:

Abstract of FR2826753

The document content is displayed on a screen of the data system., the current markup language is selected by a user on a screen display by a control device such as a pointer. The method has a stage (D) - display the list of functions applicable to the current markup language for the document. Method of processing part of an information document in a data system. The contents of the document are represented by markers, each marker has a name and a value. The value can include other markers. The document is associated with a second document, called a 'schema document' whose content is represented in a schema type markup language. The schema document defines the structure of the part of the first document. The method has the following stages: (A) - selection (E13) of a marker, called the current marker, in the part of the document; (B) - search (E15-E21) the schema document for at least a marker for a function associated with the selected language for the document; (C) - create (E23,E25) a list of applicable functions for the current marker from the function marker, found in the schema document.

Data supplied from the esp@cenet database d90 - Worldwide

Courtesy of <http://v3.espacenet.com>

12

DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 29.06.01.

30 Priorité :

43 Date de mise à la disposition du public de la
demande : 03.01.03 Bulletin 03/01.

56 Liste des documents cités dans le rapport de
recherche préliminaire : *Se reporter à la fin du
présent fascicule*

60 Références à d'autres documents nationaux
apparentés :

71 Demandeur(s) : CANON KABUSHIKI KAISHA — JP.

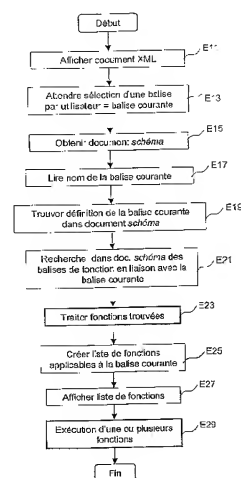
72 Inventeur(s) : MOREAU JEAN JACQUES et RUEL-
LAN HERVE.

73 Titulaire(s) :

74 Mandataire(s) : RINUY SANTARELLI.

54 PROCÉDE ET DISPOSITIF DE TRAITEMENT D'UN DOCUMENT INFORMATIQUE DANS UN SYSTEME
INFORMATIQUE.

57 Un procédé de traitement d'une portion d'un document informatique dans un système informatique, le contenu du document informatique étant représenté avec un langage de balisage, chaque balise comportant un nom et une valeur, le document informatique étant associé à un second document informatique, appelé "document schéma", dont le contenu est représenté dans un langage de balisage de type schéma, le document schéma définissant la structure de la portion considérée du document informatique. Ce procédé de traitement comporte les étapes de: sélection (E13) d'une balise, dite "balise courante", dans la portion du document informatique; recherche (E15-E21) dans le document schéma d'au moins une balise déclarative d'une fonction associée à la balise sélectionnée du document informatique; création (E23, E25) d'une liste de fonctions applicables à la balise courante du document informatique à partir d'au moins une balise déclarative d'une fonction, trouvée dans le document schéma.



5

10 La présente invention a trait de manière générale au traitement de documents informatiques dans des réseaux de communication tels que l'Internet.

Plus particulièrement, l'invention concerne un procédé de traitement d'une portion d'un document informatique dans un système informatique. Le contenu du document informatique étant représenté avec un langage de balisage, et étant associé à un second document informatique, appelé "document schéma", dont le contenu est représenté dans un langage de balisage de type schéma, ce second document définissant la structure de la portion considérée du document informatique. L'invention concerne également un système informatique apte à mettre en œuvre un tel procédé.

Le World Wide Web (Web ou WWW) constitue l'interface de prédilection pour accéder à la plupart des ressources disponibles sur Internet. Le Web (aussi appelé "Toile" en français) est un ensemble de protocoles Internet et de logiciels qui présentent l'information dans un format hypertexte. Le concept d'hypertexte s'emploie pour construire des documents informatiques qui référencent d'autres documents à l'aide de liens, aisément sélectionnables par des utilisateurs novices.

Le langage HTML (*HyperText Markup Language*) est un langage à balises conçu pour décrire la manière dont des documents doivent être affichés par les navigateurs Web. Ce langage du fait de sa simplicité et de sa facilité d'utilisation, a été rapidement adopté comme un standard dans le monde

Internet. Cependant, il est impossible d'étendre ce langage au-delà de ce pourquoi il a été conçu, c.-à-d., la présentation de documents. Les auteurs de documents ne sont en effet pas autorisés à créer de nouvelles balises pour leurs besoins spécifiques.

- 5 Le langage XML (*eXtensible Markup Language*) a été créé notamment pour pallier la faiblesse du langage HTML. Le langage XML est défini selon une norme de l'organisation internationale connue sous le nom de "W3 Consortium" (W3C).

10 XML est un langage à balises de structuration : chaque balise correspond à un élément, qui peut être présent une ou plusieurs fois dans le document. Une balise peut contenir du texte ou d'autres balises. Dans le cas ou elle contient d'autres balises, une balise est assimilée à un nœud d'un arbre. Contrairement à HTML, le nom des balises n'est pas soumis à une norme, il est défini par le programmeur. En outre, chaque balise ouvrante doit être
15 accompagnée d'une balise fermante, ou être une balise vide. Cette contrainte permet de créer des documents ayant une structure hiérarchique qui peut être représentée sous la forme d'un arbre.

 De par sa structure hiérarchique (arborescente), le langage XML permet de représenter des données structurées. D'autre part, un document XML est
20 parfaitement lisible par un humain "averti". Enfin les documents XML étant au format texte, ils peuvent être facilement échangés via le Web en utilisant les protocoles existants comme HTTP (*Hypertext Transfer Protocol*).

 Cependant, dans le langage XML, les données sont de types prédéterminés, en particulier il n'est pas possible à un programmeur de définir
25 de nouveaux types, en particulier des types complexes, à partir des types prédéterminés. Une des conséquences est que l'on ne peut pas directement définir dans un document XML des fonctions applicables à un élément ou objet représenté dans le document.

 Or, avec notamment le développement du commerce électronique sur
30 Internet, il est souhaitable de pouvoir en même temps que l'on visualise un document informatique dont le contenu est représenté en XML, directement

visualiser ou être informé des fonctions qui peuvent être lancées sur des objets contenus dans le document.

Une solution connue pour répondre à ce besoin est de fournir dans le document XML, une adresse électronique indiquant la localisation du code exécutable d'une fonction applicable à un objet du document. Cette solution présente l'inconvénient de nécessiter une opération supplémentaire de chargement d'un autre document contenant le code exécutable représenté dans un autre langage pour pouvoir exécuter une fonction. D'autre part, pour pouvoir exécuter sur un ordinateur local la fonction appelée, il est nécessaire de disposer sur cet ordinateur d'un logiciel spécifique pour interpréter ou compiler le code exécutable de la fonction et pouvoir ainsi l'exécuter.

En outre une fonction n'est utilisable qu'en relation avec un seul objet, sans possibilité de s'appliquer à un autre objet avec des paramètres d'entrée (arguments) de la fonction différents.

En complément du langage XML, on trouve des langages "de schéma", ainsi désignés parce qu'ils permettent de définir une classe de document XML, le terme "document instance" étant souvent utilisé pour définir un document qui est valide par rapport à un certain schéma. Ainsi une DTD (*Document Type Definition*) est un langage de schéma qui permet de spécifier quels éléments sont susceptibles d'apparaître dans les documents instances, quels éléments peuvent être contenus dans d'autres documents, et dans quel ordre ils doivent apparaître. Le plus souvent une DTD est un fichier à part, externe au document XML.

Récemment, un autre langage de schéma est apparu : XML Schema, également défini par une norme W3C. XML Schema présente l'avantage par rapport aux DTD d'utiliser la syntaxe XML. Autrement dit, un document exprimé en XML Schema (appelé schéma XML) est un document XML. Ainsi n'importe quel schéma XML peut être manipulé par tout éditeur XML.

Le langage XML Schema présente, entre autres, l'avantage par rapport au langage XML, de permettre la création de types complexes.

La présente invention a pour objet d'utiliser les possibilités de création de types complexes offertes par un langage de schéma tel que le langage XML Schema, dans le but de définir simplement des fonctions exécutables sur un objet contenu dans un document XML, et d'associer ces fonctions
5 automatiquement au document XML, lorsque celui-ci est présenté à un utilisateur par exemple via un écran d'ordinateur, ou bien lorsque le document est directement traité par une application informatique.

A cet effet, la présente invention concerne, selon un premier aspect, un procédé de traitement d'une portion d'un document informatique dans un
10 système informatique, le contenu du document informatique étant représenté avec un langage de balisage, chaque balise comportant un nom et une valeur, ladite valeur pouvant inclure d'autres balises; le document informatique étant associé à un second document informatique, appelé "document schéma", dont le contenu est représenté dans un langage de balisage de type schéma, le
15 document schéma définissant la structure de la portion considérée du document informatique.

Conformément à l'invention, le procédé de traitement comporte les étapes suivantes :

- sélection d'une balise, dite "balise courante", dans la portion du
20 document informatique ;
- recherche dans le document schéma d'au moins une balise déclarative d'une fonction associée à la balise sélectionnée du document informatique ;
- création d'une liste de fonctions applicables à la balise courante du document informatique à partir d'au moins une balise déclarative d'une
25 fonction, trouvée dans le document schéma.

Dans le mode de réalisation préféré, le document informatique est un document XML, et le document schéma est un schéma XML. L'invention permet ainsi d'associer automatiquement une liste de fonctions à une balise sélectionnée d'un document XML. Cette liste est créée automatiquement lors
30 de l'utilisation du document XML, par exemple sa visualisation par un utilisateur humain par l'intermédiaire d'un écran d'ordinateur.

Selon un mode préféré de réalisation, le contenu du document informatique est affiché sur un écran d'affichage du système informatique, la balise courante étant sélectionnée par un utilisateur sur l'écran d'affichage au moyen d'un dispositif de commande, par exemple un dispositif de pointage. Le
5 procédé comporte alors une étape d'affichage de la liste de fonctions applicables à la balise courante dudit document informatique.

De cette façon, un utilisateur consultant un document XML, peut par exemple, par le simple clic d'une souris pointant sur une balise choisie d'un document XML en cours de visualisation, être informé immédiatement des
10 fonctions exécutables qui s'appliquent à cette balise.

Selon le mode de réalisation préféré, la structure de chacune des balises contenues dans la portion du document informatique est définie dans le document schéma par des balises appropriées dans lesquelles un champ particulier contient le nom de la balise correspondante du document
15 informatique. Ainsi, l'étape d'identification d'au moins une balise déclarative d'une fonction associée à la balise courante dans le document informatique, est effectuée rapidement par, dans un premier temps, la recherche dans le document schéma d'une balise comportant le nom de la balise courante, et dans un second temps, la recherche de balises déclaratives de fonction, en
20 liaison avec la balise trouvée précédemment.

Selon une caractéristique particulière de l'invention, l'étape de création d'une liste de fonctions applicables à la balise sélectionnée (la balise courante) du document informatique, inclut, pour chaque balise déclarative d'une fonction trouvée dans le document schéma, les sous-étapes suivantes :

- 25 - lecture d'au moins une balise déclarative d'un argument de la fonction, ladite balise comportant un champ définissant le nom de l'argument, et un champ définissant le type de l'argument ;
- détermination de la valeur de chaque argument par affectation audit argument de la valeur d'une balise du document informatique qui est identifiée
30 selon un mode d'identification codé dans la balise déclarative de l'argument, dans le document schéma, ou bien par affectation audit argument d'une valeur

fournie par l'utilisateur au moyen d'un dispositif de commande tel qu'un clavier d'ordinateur.

On peut alors faire aisément et directement des appels de fonctions à partir d'une balise courante d'un document informatique, puisque l'on dispose
5 d'une liste de fonctions instanciées c.-à-d. dont les arguments ont une valeur déterminée.

Selon une autre caractéristique préférée, le procédé de traitement selon l'invention comporte en outre une étape d'exécution d'au moins une fonction de la liste affichée, en réponse à une commande générée par une action de
10 l'utilisateur sur un dispositif de commande tel qu'un clavier d'ordinateur ou un dispositif de pointage.

Un utilisateur dispose ainsi d'un moyen efficace et simple d'emploi pour initier des traitements sur un document informatique (par ex. un document XML).

15 Dans un mode de réalisation dans lequel le système informatique est relié à un réseau de communication, par exemple, l'Internet, la fonction choisie par l'utilisateur peut être exécutée à distance sur un second système informatique relié au premier système informatique via le réseau.

En pratique, dans le document schéma, chaque balise déclarative d'une
20 fonction inclut en outre l'adresse électronique du code exécutable de la fonction, ou bien inclut directement le code exécutable de la fonction, le code exécutable pouvant être sous forme de code source, code objet, ou de code intermédiaire entre code source et code objet.

Il n'est pas nécessaire, par conséquent, d'installer le code de la fonction
25 sur un poste client relié à un réseau de communication. D'autre part, quelque soit la nature du code de la fonction, code source ou code intermédiaire entre code source et code compilé, il n'est pas nécessaire de pré-compiler, dans un poste serveur, un code de la fonction spécifique à chaque système d'exploitation (OS) différent équipant les postes clients sur le réseau.

30 Selon une variante de réalisation ou bien de manière cumulative avec le procédé tel que brièvement décrit ci-dessus, un procédé de traitement d'un

document informatique selon l'invention peut également être défini par les étapes principales suivantes :

- sélection d'une balise, dite "balise courante", dans ladite portion du document informatique ;

5 - recherche dans le document schéma d'une balise déclarative d'une fonction prédéterminée, associée à la balise sélectionnée du document informatique ;

- exécution de ladite fonction prédéterminée.

10 Par exemple, le nom de la fonction à exécuter pourra être connu d'une application informatique, qui assurera automatiquement la recherche de la fonction dans le document schéma, puis lancera automatiquement l'exécution de la fonction, sur le même système informatique traitant le document informatique, ou bien à distance, sur un autre système informatique, au travers d'un réseau de communication.

15 Selon un deuxième aspect, l'invention concerne un système informatique, relié ou non à un réseau de communication, caractérisé en ce qu'il comporte des moyens adaptés à la mise en œuvre d'un procédé de traitement d'un document informatique, tel que brièvement exposé supra.

20 La présente invention concerne encore un programme d'ordinateur sur un support d'informations. Ce programme d'ordinateur étant remarquable en ce qu'il comporte des instructions de programme adaptées à la mise en œuvre d'un procédé de traitement d'un document informatique tel que brièvement exposé supra, lorsque ce programme d'ordinateur est chargé et exécuté dans un ordinateur.

25 L'invention vise aussi un support d'informations contenant un tel programme d'ordinateur. Un tel support d'informations peut comporter un moyen de mémorisation, tel qu'une ROM, par exemple un CD ROM ou une ROM semi-conducteur, ou un moyen d'enregistrement magnétique, par exemple une disquette (*floppy disc*) ou un disque dur. D'autre part, le support
30 peut être un support transmissible tel qu'un signal électrique ou optique, qui

peut être acheminé via un câble électrique ou optique, ou encore par radio ou par d'autres moyens.

Les avantages de ce dispositif, programme d'ordinateur, et de ce support d'informations, sont identiques à ceux du procédé de traitement d'un document
5 informatique en conformité avec l'invention, tels que brièvement exposés supra.

D'autres particularités et avantages de l'invention apparaîtront encore dans la description ci-après d'un mode préféré de réalisation de la présente invention décrit à l'appui des dessins annexés sur lesquels :

- la FIG. 1 est un organigramme illustrant un procédé de traitement d'un
10 document informatique selon l'invention ;

- la FIG. 2 est un organigramme détaillant l'étape de traitement (E23), représentée à la FIG. 1, des fonctions associées à une balise contenue dans un document informatique ;

- les FIGs. 3a et 3b détaillent l'étape d'obtention (E233) de la valeur d'un
15 argument représentée à la FIG. 2 ;

- la FIG. 4 est un organigramme détaillant l'étape, représentée à la FIG. 1 (E29), d'exécution de fonctions associées à une balise contenue dans un document informatique ;

- la FIG. 5 représente un système informatique apte à mettre en œuvre
20 un procédé de traitement d'un document informatique selon l'invention.

Dans la description qui suit, on va décrire un procédé de traitement d'un document informatique selon l'invention.

Plus précisément, il s'agit d'un procédé de traitement d'une portion d'un document informatique dans un système informatique. Le contenu du
25 document informatique est représenté avec un langage de balisage dans lequel chaque balise comporte un nom et une valeur. La valeur d'une balise pouvant également contenir d'autres balises.

Ce document informatique est associé à un second document informatique, appelé "document schéma", dont le contenu est représenté dans
30 un langage de balisage de type schéma (comme exposé plus haut). Ce

document schéma définit la structure de la portion à traiter du document informatique.

Dans le cadre de l'exemple d'implémentation illustré et décrit ci-après, le document informatique à traiter est un document dont le contenu est représenté en langage XML. Tandis que le document de schéma est un document dont le contenu est représenté en langage XML *Schema*.

On donne ci-dessous un exemple de portion d'un document XML (livre.xml) traitée selon l'invention dans un système informatique tel qu'un micro-ordinateur.

10

Document informatique XML :

```
<livre>
  <titre>Le Rouge et le Noir</titre>
  <auteur>Stendhal</auteur>
  <personnages>
    <personnage>Julien Sorel</personnage>
    <personnage>Mathilde de la Mole</personnage>
    <personnage>Mme de Reynal</personnage>
    <personnage>M. de Reynal</personnage>
    ...
  </personnages>
</livre>
```

Cette portion de document XML, décrit un élément principal, un livre (balise "livre"), comportant des sous-balises (encore appelées "balise fille") : "titre", "auteur", "personnages". Le contenu de la balise "personnages" est constituées de plusieurs sous-balises, chacune desquelles, contenant le nom d'un seul personnage du livre.

Ce document XML, peut être par exemple obtenu sur Internet par un utilisateur cliquant avec une souris sur un objet informatique publié par un site Web.

Conformément à l'état de la technique à la date de dépôt de la présente demande, un schéma XML, appelé "livre.xsd" définissant la structure du document XML précédent (livre.xml) est donné ci-dessous à titre d'exemple.

5 Document schéma "livre.xsd" selon l'état de la technique :

```

<xs:element name="livre">
  <xs:complexType>
    <xs:sequence>
10      <xs:element name="titre" type="xs:string"/>
        <xs:element name="auteur" type="xs:string"/>
        <xs:element name="personnages">
          <xs:complexType>
            <xs:sequence maxOccurs="unbounded">
15              <xs:element name="personnage" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
20    </xs:complexType>
  </xs:element>

```

Comme on peut le voir dans le document schéma ci-dessus, l'élément livre est défini comme un élément de type complexe ("*xs:complexType*")
 25 comprenant plusieurs sous-éléments (c.-à-d. des sous-balises). Ces sous-éléments sont les suivants :

- élément "titre" de type "chaîne de caractères" (*string*),
- élément "auteur" de type "string",
- élément "personnages" de type complexe (*complexType*), comportant
 30 une séquence de sous-éléments : "personnage" de type "string", en nombre maximum non borné ("*unbounded*").

Ainsi chaque balise du document XML Schema (livre.xsd), identifiée chacune par son nom (champ "*name*"), définit la structure d'une balise correspondante, portant le même nom, dans le document XML (livre.xml).

Comme on peut le voir dans le document schéma ci-dessus, il n'y a aucune balise permettant de déclarer des fonctions associés à des éléments du document XML.

Conformément à l'invention, on déclare une balise déclarative de fonction dans le document schéma XML de la façon suivante :

```
<xs:function name="..." text="..."/>
```

10 où :

"function" est le mot clé identifiant une balise de déclaration de fonction,

"name=" est un mot clé identifiant un champ dont la valeur définit le nom de la fonction,

15 "text=" est un mot clé identifiant un champ optionnel dont la valeur est du texte destiné à être affiché pour l'utilisateur, lors de l'affichage de la liste de fonctions.

D'autre part, à l'intérieur de chacune des balises de fonctions telles que définies ci-dessus dans le document schéma, on exprime conformément à l'invention des balises déclaratives d'un argument de la façon suivante :

```
<xs:argument name="..." type="..."/>
```

...

```
<xs:argument name="..." type="..."/>
```

où :

25 "argument" est le mot clé identifiant une balise de déclaration d'un argument,

"name=" est un mot clé identifiant un champ dont la valeur définit le nom de l'argument,

30 "type=" est un mot clé identifiant un champ dont la valeur définit le type de l'argument.

On donne ci-dessous, un exemple de schéma XML conforme à l'invention, cet exemple est issu du schéma XML "livre.xsd" précédent, modifié selon l'invention :

5 Document schéma "livre.xsd" modifié selon l'invention :

```

<xs:element name="livre">
  <xs:complexType>
    <xs:function name="purchase" text="Acheter ce livre"
10      href="http://oceania/web-obj/obj/purchase.exe"/>
    <xs:arguments>
      <xs:argument name="auteur" type="xs:string"/>
      <xs:argument name="titre" type="xs:string"/>
    </xs:arguments>
15  </xs:function>
    <xs:function name="orderTeaser" text="Télécharger le 1er chapitre"
      href="http://oceania/web-obj/obj/orderTeaser.exe"/>
    <xs:arguments>
      <xs:argument name="auteur" type="xs:string"/>
20      <xs:argument name="titre" type="xs:string"/>
    </xs:arguments>
    </xs:function>
    <xs:element name="titre" type="xs:string"/>
    <xs:element name="auteur" type="xs:string"/>
25  <xs:element name="personnages">
    <xs:complexType>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="personnage" type="xs:string"/>
        <xs:complexType>
30          <xs:function name="viewPicture" text="Voir la photo"
            href="http://oceania/web-obj/obj/viewPicture.exe"/>
          <xs:arguments>
            <xs:argument

```

```

        name="auteur"
        fromTag="../auteur"
        type="xs:string"/>
5      <xs:argument
        name="titre"
        fromTag="../titre"
        type="xs:string"/>
      <xs:argument
10      name="personnage"
        fromTag="personnage"
        type="xs:string"/>
    </xs:arguments>
  </xs:function>
  <xs:function name="playText" text="Entendre le texte"
15      href="http://oceania/web-obj/obj/playText.exe"/>
    <xs:arguments>
      <xs:argument
        name="auteur"
        fromTag="../auteur"
20      type="xs:string"/>
      <xs:argument
        name="titre"
        fromTag="../titre"
        type="xs:string"/>
25      <xs:argument
        name="personnage"
        fromTag="personnage"
        type="xs:string"/>
    </xs:arguments>
30  </xs:function>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
```

```

    </xs:element>
  </xs:complexType>
</xs:element>

```

5 Comme on peut le voir, la structure de chacune des balises "<nom>...</nom>" (par ex. "<titre>...</titre>") contenues dans la portion du document XML considérée, est définie dans le document schéma par des balises appropriées :

```
<xs:element name="..." .../>
```

10 ou bien

```
<xs:element name="..." ...> ... </xs:element>
```

dans lesquelles le champ particulier, name="...", contient le nom de la balise correspondante du document XML.

Dans l'exemple ci-dessus, les éléments supplémentaires conformes à
15 l'invention sont représentés en gras. Ainsi le document schéma "livre.xsd" selon l'invention comporte plusieurs fonction déclarées en liaison avec la balise de nom "livre" (name="livre") :

- fonction nommée "purchase" (acheter en anglais), permettant, lorsqu'elle est exécutée, d'acheter le livre ;

20 - fonction nommée "orderTeaser" permettant, lorsqu'elle est exécutée, de télécharger le premier chapitre du livre.

Ces fonctions associées dans le document schéma à la balise de nom "livre" (name="livre") sont ainsi directement associées à la balise "livre" du document XML (c.-à-d. la balise "<livre> ... </livre>").

25 On notera ici que chacune de ces balises de fonction comporte un attribut text="...", permettant d'afficher sur un écran d'ordinateur à destination d'un utilisateur humain la chaîne de caractères codée dans ce champ (par ex., "Acheter ce livre").

De même, associées dans le document schéma à la balise de nom
30 "personnage" se trouvent déclarées deux fonctions :

- fonction nommée "viewPicture" permettant, lorsqu'elle est exécutée, de voir une photo du personnage sélectionné ;

- fonction nommée "playText" permettant, lorsqu'elle est exécutée, d'entendre un enregistrement de la voix d'un acteur de théâtre interprétant le personnage sélectionné et récitant un morceau de texte.

Comme on peut le voir dans l'exemple de code XML Schema ci-dessus, chacune des fonctions comportent des arguments déclarés par des balises : `<xs:argument name="..." type="..." />`.

Ainsi la fonction "viewPicture" comporte trois arguments :

- argument "auteur" (name="auteur"),
- argument "titre" (name="titre"),
- argument "personnage" (name="personnage").

Dans cet exemple, lorsqu'une fonction comporte plus d'un argument, les balises déclarant ces arguments sont "encadrées" par deux balises de déclaration de liste d'arguments : balise `<xs:arguments> ... </xs:arguments>`.

Conformément à l'invention, chaque balise déclarative d'une fonction, dans le document XML Schema, inclut en outre l'adresse électronique du code exécutable de la fonction.

En pratique, cette adresse est fournie dans un champ particulier correspondant à l'attribut "href" (href="..."). Ainsi, la fonction "playText" comporte l'attribut : href="http://oceania/web-obj/obj/playText.exe" signifiant que le code exécutable de la fonction "playText" se trouve sur un autre système informatique, le serveur "oceania", dans le répertoire "web-obj/obj/" et, dans ce répertoire, dans le fichier "playText.exe". Bien sûr, dans cet exemple le système informatique considéré est relié à un réseau de type Internet, l'adresse électronique de la fonction étant une adresse HTTP. Dans ce cas, la fonction est alors exécutable à distance sur le réseau.

Cependant, en variante, on peut inclure directement le code exécutable de la fonction dans le document schéma. Par ailleurs, ce code exécutable peut être sous forme de code source, code objet, ou de code intermédiaire entre code source et code objet.

Comme mentionné précédemment, chaque fonction déclarée dans le document XML Schema, comporte une déclaration d'un ou plusieurs arguments définis par une balise du type : `<xs:argument name="..." type="..."/>`.

5 Comme on peut le voir sur l'exemple de document schéma ci-dessus, certaines balises d'arguments, par exemple celles associées à la fonction "playText", comporte un attribut supplémentaire : `fromTag="..."`.

Conformément à l'invention, la valeur de cet attribut (champ "...") est représentative d'un chemin dans le document XML permettant de trouver la
10 balise dont la valeur est à affecter à l'argument considéré. Cela sera détaillé plus loin dans la description, en liaison avec les FIGs. 3a et 3b.

En liaison avec la **FIG. 1** on va à présent décrire un procédé de traitement d'un document informatique selon l'invention, ce procédé utilisant les possibilités nouvelles offertes par l'association à un document informatique
15 (XML) d'un document schéma tel que décrit ci-dessus.

Comme illustré à la FIG. 1, le procédé de traitement commence par une étape d'affichage E11 du document XML sur un écran d'affichage associé au système informatique considéré, par ex. un micro-ordinateur. Dans une implémentation préférée, le système informatique est relié au réseau Internet,
20 et le document XML est affiché par l'utilisation d'un logiciel de navigation Internet (*Web Browser* en anglais).

A l'étape E13 qui suit on attend la sélection d'une balise par un utilisateur, par ex. au moyen d'un dispositif de commande, par ex. un dispositif de pointage tel qu'une souris. L'utilisateur pourra, par exemple, positionner le
25 curseur directement sur une balise du document XML. La balise sélectionnée devient alors la "balise courante" du document XML.

Dans le mode de réalisation choisi et représenté à la FIG. 1, la sélection d'une balise du document XML est effectuée par un utilisateur, cependant dans un autre mode de réalisation cette sélection peut être effectuée
30 automatiquement par une application logicielle auquel le nom de la balise à sélectionner aura été préalablement fourni comme paramètre d'entrée.

Ensuite, à l'étape E15, le document schéma associé au document XML est automatiquement chargé sur l'ordinateur, l'adresse du document schéma est mentionné dans une partie prologue du document XML. Cette adresse peut être locale, ou distante, c.-à-d., pointant sur un autre système informatique accessible via le réseau.

Lorsque le document schéma est chargé, on lit le nom de la balise courante (étape E17) (par ex. "livre"). Ensuite, la définition (c.-à-d. la structure) de la balise courante est recherchée dans le document schéma (étape E19). L'implémentation de la mise en correspondance d'une balise contenue dans le document XML et d'une balise du document schéma qui la définit, est en soi connue de l'homme du métier. Expliqué de façon simplifiée, pour effectuer cette mise en correspondance le logiciel peut rechercher, dans le document schéma, la balise dont l'attribut "name" contient les mêmes caractères que le nom de la balise courante, dans le document XML.

Ainsi, si la balise courante est : `<personnage>Julien Sorel</personnage>`, la balise `<xs:element name="personnage" type="xs:string"/>` est trouvée dans le document schéma.

Ensuite, à l'étape E21, on recherche dans le document schéma les balises déclaratives d'une fonction qui sont associées à la balise courante (sélectionnée à l'étape E13) du document informatique.

En pratique, on procède de la façon suivante : on recherche dans le document schéma une balise comportant le nom de la balise courante, puis on recherche des balises déclaratives de fonction en liaison avec la balise trouvée qui comporte le nom de la balise courante.

Ainsi, à titre d'exemple, si la balise courante est : `<personnage>Julien Sorel</personnage>`, la balise trouvée dans le document schéma, à l'étape E19, est : `<xs:element name="personnage" type="xs:string"/>`. Les balises déclaratives de fonction trouvées à l'étape E21 sont les suivantes :

`<xs:function name="viewPicture" text="Voir la photo"`
`href="http://oceania/web-obj/obj/viewPicture.exe"/>`

et

```
<xs:function name="playText" text="Entendre le texte"
  href="http://oceania/web-obj/obj/playText.exe"/>
```

A l'étape suivante, E23, on procède au traitement des fonctions trouvées.

- 5 Plus précisément, il s'agit de donner des valeurs aux arguments définis pour chacune des fonctions. Cette étape sera détaillée plus bas en liaison avec la FIG. 2.

- A l'étape qui suit, E25, on crée la liste des fonctions applicables à la balise courante du document XML, à partir des balises déclaratives de fonction
10 trouvées et traitées, respectivement aux étapes précédentes E21 et E23. Cette liste est alors mémorisée temporairement, localement sur l'ordinateur.

- Dans le mode de réalisation préféré représenté à la FIG.1, dans lequel le contenu du document XML est affiché sur un écran du système informatique pour être utilisé par un utilisateur, le procédé de traitement d'un document
15 informatique selon l'invention, comporte une étape (E27) d'affichage de la liste de fonctions applicables à la balise sélectionnée (balise courante) du document XML.

- Enfin, à l'étape E29, au moins une fonction de la liste de fonctions affichée, est exécutée, en réponse à une commande générée par une action de
20 l'utilisateur sur un dispositif de commande tel qu'un clavier d'ordinateur ou un dispositif de pointage.

- En pratique, une nouvelle fenêtre est affichée à l'écran montrant les fonctions applicables à la balise sélectionnée du document XML. Par exemple, si la balise sélectionnée est : <livre> dans le document XML donné en exemple,
25 il sera affiché à l'écran, le texte qui suit, indiquant à l'utilisateur la fonction applicable directement à la balise <livre> : Acheter ce livre

Le code XML correspondant à cette fonction qui sera alors incorporé dans une requête d'exécution de la fonction est le suivant :

- 30 <purchase>
 <titre>Le Rouge et Le Noir</titre>

<auteur>Stendhal</auteur>
</purchase>

Le processus d'exécution d'une ou plusieurs fonctions (étape E29) sera
 5 détaillé plus loin en liaison avec la FIG. 4.

Dans un autre mode de réalisation, dans lequel la sélection d'une balise
 du document XML est automatiquement réalisée par une application logicielle
 auquel le nom de la balise à sélectionner aura été préalablement fourni comme
 paramètre d'entrée, à l'étape E21, il est procédé à la recherche, dans le
 10 document schéma, d'une balise déclarative d'une fonction prédéterminée,
 associée à la balise sélectionnée du document informatique. Le nom de cette
 fonction prédéterminée est fournie par exemple comme un paramètre d'entrée
 à l'application logicielle.

Dans ce mode de réalisation, les étapes de création et d'affichage (E25,
 15 E27) d'une liste de fonctions ne sont pas effectuées.

Enfin, dans ce mode de réalisation, à l'étape E29, la fonction
 prédéterminée est automatiquement exécutée, sans intervention d'un
 utilisateur.

En liaison avec la **FIG. 2**, on va maintenant détailler l'étape de traitement
 20 (E23), représentée à la FIG. 1, des fonctions associées à une balise
 sélectionnée dans le document XML.

Le processus de traitement illustré à la FIG. 2 est exécuté pour chaque
 fonction identifiée (E21) dans le document schéma, en liaison avec la balise
 courante du document XML.

25 On commence par une étape de sélection (E230) d'une première balise
 de fonction dans le document schéma. Par ex. si la balise courante dans le
 document XML est : *<personnage>Julien Sorel</personnage>*, la balise
 trouvée dans le document schéma, à l'étape E19 (FIG. 1), est : *<xs:element*
name="personnage" type="xs:string"/>. Les balises déclaratives de fonction
 30 trouvées à l'étape E21 sont les suivantes :

<xs:function name="viewPicture" text="Voir la photo"

href="http://oceania/web-obj/obj/viewPicture.exe"/>

et

<xs:function name="playText" text="Entendre le texte"

href="http://oceania/web-obj/obj/playText.exe"/>

5 Ainsi, dans cet exemple, la balise relative à la fonction "viewPicture" est d'abord sélectionnée.

Ensuite à l'étape E231, on sélectionne une balise d'argument relative à la fonction en cours. Par exemple, dans le cas de la fonction "viewPicture" les balises d'arguments sont les suivantes :

10 *<xs:argument name="auteur" fromTag=".././auteur" type="xs:string"/>*

<xs:argument name="titre" fromTag=".././titre" type="xs:string"/>

<xs:argument name="personnage" fromTag="personnage" type="xs:string"/>

Ainsi, on commence par exemple par la balise d'argument "auteur".

15 Puis, on lit (E232) dans la balise d'argument sélectionnée, le nom de l'argument (attribut : name="...") et le chemin, ou plus généralement le mode d'accès, pour accéder à la valeur de l'argument dans le document XML. Ce chemin est codé par la valeur de l'attribut "fromTag". Pour la balise "auteur" ce chemin est: *".././auteur"*.

20 A l'étape E233, la valeur de l'argument considéré est obtenue dans le document XML selon la valeur de l'attribut "fromTag". La valeur de l'attribut "fromTag" définit ainsi un mode d'identification de la valeur d'une balise du document XML à affecter à l'argument considéré.

Le mode de détermination de la valeur d'un argument (E233), par définition d'un chemin, sera détaillé plus loin en liaison avec les FIGs. 3a-3b.

25 Alternativement, la valeur d'un argument peut également être obtenue par identification dans le document XML de la balise dont le nom est identique au nom défini dans la balise déclarative de l'argument dans le document schéma.

30 C'est par exemple le cas pour la balise d'argument : *<xs:argument name="auteur" type="xs:string"/>* associée à la fonction "purchase" (dans le document schéma) qui est définie pour la balise *<livre>* du document XML.

Ainsi la valeur de l'argument "auteur" de la fonction "purchase" correspond à la valeur de la balise <auteur> dans le document XML, c.-à-d. la chaîne de caractères : "Stendhal".

5 Selon une variante de réalisation complémentaire ou alternative aux deux qui précèdent, la valeur à affecter à un argument peut être fournie par un utilisateur au moyen d'un dispositif de commande tel qu'un clavier d'ordinateur.

De retour à la FIG. 2, à l'étape E234, on teste si l'argument dont la valeur a été déterminée est le dernier ou non des arguments définis pour la fonction en cours. Si ce n'est pas le cas, on retourne à l'étape E231 pour sélectionner
10 une autre balise d'argument, et le processus recommence comme précédemment exposé.

Au contraire, si c'est le cas, on passe à l'étape E235 à laquelle on procède à la sauvegarde en mémoire de la fonction ainsi traitée (c.-à-d. fonction et valeur des arguments correspondants).

15 De même, à l'étape E236 on teste si la fonction qui vient d'être mémorisée est la dernière ou non des fonctions déclarées dans le document schéma pour la balise courante dans le document XML. Dans la négative, on retourne à l'étape E230 et le processus recommence comme exposé précédemment.

20 Dans l'affirmative, l'étape de traitement (E23) des fonctions correspondant à une balise sélectionnée dans le document XML (balise courante) est terminée.

En liaison avec les **FIGs. 3a-3b**, on va à présent détailler l'étape de détermination (E233, FIG. 2) de la valeur d'un argument, dans le cas où le
25 mode de détermination de cette valeur est codé par l'attribut "fromTag" d'une balise d'argument dans le document XML Schema.

Conformément à l'invention, comme exposé ci-après, le mode d'identification d'une balise du document XML dont la valeur est à affecter à un argument considéré, est codé dans la balise déclarative de cet argument,
30 comme une information définissant un chemin dans le document XML, à partir de la balise courante, jusqu'à la balise dont la valeur est à affecter à l'argument.

A la **FIG. 3a**, on commence par une étape de test E2330 à laquelle il est déterminé si l'information codée dans l'attribut "fromTag" de la balise d'argument considérée définit ou non un chemin relatif.

5 Cas 1 : Une première forme de chemin relatif est définie selon l'invention par la syntaxe : *fromTag=".././nom_de_balise"* où "*nom_de_balise*" désigne le nom d'une balise du document XML. Par exemple dans le document schéma on a : *fromTag=".././auteur"* dans la balise d'argument "auteur" associée à la fonction "playText" dans le document schéma.

10 Cas 2 : Une seconde forme de chemin relatif est selon l'invention définie par la syntaxe : *fromTag="nom_de_balise"*, par exemple, *fromTag="personnage"* comme c'est le cas dans la balise d'argument "personnage" associée à la fonction "playText" dans le document schéma.

15 Cas 3 : Un chemin non relatif est défini selon l'invention par la syntaxe *fromTag="/nom_de_balise"*, par exemple, *fromtag="/auteur"* (non présent dans l'exemple de document schéma donné ici).

Ainsi à l'étape E2330, si le chemin n'est pas un chemin relatif, par ex. *fromTag="/auteur"* (Cas 3 ci-dessus), à l'étape E2332, le programme selon l'invention pointe sur la balise de début (la balise racine) du document XML traité. Dans l'exemple donné ici, il s'agit de la balise : <livre>.

20 Ensuite à l'étape E2333, on applique le chemin défini par l'attribut *fromTag* au document XML. Dans le cas du chemin *"/auteur"* on pointe alors sur la balise fille <auteur> du document XML. Ensuite à l'étape E2334, on lit la valeur de la balise pointée à l'étape précédente, c.-à-d. : "Stendhal".

25 Enfin, à l'étape E2335, on renvoie cette valeur pour l'affecter à l'argument considéré de la fonction traitée dans le document schéma.

De retour à l'étape E2330, si le chemin codé par l'attribut *fromTag* est un chemin relatif (Cas 1 ou 2 supra), à l'étape E2331, on pointe sur la balise courante du document XML.

30 Ainsi, dans le Cas 2 supra, si la balise courante est :
 <personnage>Julien Sorel</personnage>

la valeur de l'attribut "fromTag" étant "*personnage*", à l'étape E2333 on reste sur cette balise et on lit à l'étape E2334 sa valeur : "Julien Sorel" que l'on envoie à l'argument considéré de la fonction traitée (par ex. "playText").

Dans l'exemple donné plus haut avec le Cas 1, la balise courante
5 pointée à l'étape E2331 serait par ex. : *<personnage>Julien Sorel</personnage>*

Puis, à l'étape E2333, on parcourt dans le document XML le chemin défini par la valeur de l'attribut "fromTag", soit : "*.././auteur*".

En conséquence, on remonte d'abord d'un rang selon l'instruction : "*../*"
10 dans la structure d'arbre définie par le document XML, et on pointe alors sur la balise *<personnages>*. Ensuite on remonte encore d'un rang selon la seconde instruction "*../*" et on aboutit alors à la balise *<livre>*. Finalement, selon l'instruction "*/auteur*", on pointe sur la balise : *<auteur>Stendhal</auteur>*.

Enfin on lit la valeur de la balise pointée c.-à-d., "Stendhal", à l'étape
15 E2334, puis on renvoie (E2335) cette valeur à la balise d'argument considérée (par ex. argument "auteur") de la fonction traitée (par ex. "playText") dans le document schéma.

La **FIG. 3b** illustre le traitement selon deux modes de définition de chemin codés dans une balise d'argument selon l'invention. Ces deux modes
20 correspondent respectivement au mode relatif selon le Cas 1 supra, et au mode non relatif selon le Cas 3 supra.

Le schéma de la FIG. 3b illustre la structure arborescente du document XML (livre.xml) exposé plus haut. Le parcours fléché 31 indique un chemin codé par : *fromTag=".././auteur"* (Cas 1 supra). Comme illustré, selon le
25 parcours 31, on remonte d'un rang (*../*) à partir de la balise courante "personnage" de valeur "Julien Sorel", on aboutit alors à la balise "personnages", puis on remonte encore d'un rang (*..*), on aboutit alors à la balise "livre" ; enfin on descend d'un rang (*/auteur*) pour terminer sur la balise "auteur".

30 D'autre part, le parcours fléché 30 est relatif à un chemin codé par : *fromTag="/auteur"* (Cas 3 supra). La balise courante est également la balise

"personnage" de valeur "Julien Sorel". Comme exposé plus haut, on commence par remonter à la balise racine ("livre") et on descend vers la balise indiquée ("auteur").

En liaison avec la **FIG. 4**, on va maintenant détailler l'étape d'exécution d'une fonction applicable à une balise d'un document XML (étape E29, FIG. 1). Comme mentionné plus haut en liaison avec la FIG. 1, une nouvelle fenêtre est affichée à l'écran montrant les fonctions applicables à la balise sélectionnée du document XML. Par exemple, si la balise sélectionnée dans le document XML donné en exemple est : `<personnage>Julien Sorel</personnage>`,
il sera affiché à l'écran, pour indiquer à l'utilisateur les fonctions qui s'appliquent directement à la balise courante, le texte suivant :

Voir la photo

Entendre la voix

Le code XML correspondant à la première fonction ("voir la photo") qui sera alors incorporé dans une requête d'exécution de cette fonction est le suivant :

```
<viewPicture>
    <auteur>Stendhal</auteur>
    <titre>Le Rouge et Le Noir</titre>
    <personnage>Julien Sorel</personnage>
</viewPicture>
```

Le code XML correspondant à la seconde fonction ("entendre la voix") qui sera alors incorporé dans une requête d'exécution de cette fonction est le suivant :

```
<playText>
    <auteur>Stendhal</auteur>
    <titre>Le Rouge et Le Noir</titre>
```

<personnage>Julien Sore</personnage>
</PlayText>

Le processus d'exécution d'une fonction illustré à la FIG. 4, commence
5 par l'étape E290 à laquelle le programme de l'invention se place en mode
d'attente de la sélection par un utilisateur d'au moins une fonction parmi la liste
affichée. Dans l'exemple ci-dessus, l'utilisateur pourra par exemple cliquer sur
"Voir la photo" et/ou "Entendre la voix".

Bien sûr, dans l'autre mode de réalisation dans lequel une (ou plusieurs)
10 fonction prédéterminée est automatiquement exécutée par l'application
logicielle selon l'invention, cette étape d'attente d'une sélection par un
utilisateur n'est pas effectuée.

A l'étape suivante, on détermine pour chaque fonction sélectionnée, si la
fonction est locale, c.-à-d. si le code exécutable de la fonction est stocké sur
15 l'ordinateur utilisateur ou bien si le code exécutable est accessible par
téléchargement depuis un système informatique serveur sur le réseau. Dans ce
but, la valeur de l'attribut "href" associé à chaque fonction dans le document
schéma est analysé. Dans l'exemple donné plus haut (livre.xsd), toutes les
fonctions définies dans le document schéma sont exécutables à distance sur
20 un serveur "oceania".

Cependant, si la fonction est locale ou bien si le code exécutable de la
fonction a déjà été chargé précédemment dans l'ordinateur client, l'exécution
de la fonction est immédiate (étape E292). Dans le cas contraire, un message
de requête d'exécution à distance est généré et envoyé au travers du réseau à
25 destination du système serveur (étape E293). En pratique, il s'agit d'une
requête HTTP.

A l'étape E294, on se place alors dans un état d'attente d'un message de
réponse (réponse HTTP) contenant soit un résultat (par ex. image numérique
ou voix numérisée) et/ou un message indiquant que l'exécution de la fonction
30 s'est bien terminée ou non. Enfin à l'étape E295, le résultat de l'exécution de la

fonction est affiché, par exemple à l'intérieur d'une nouvelle fenêtre affichée à l'écran, et/ou utilisé, sur le système utilisateur.

Ainsi dans l'exemple ci-dessus, si l'utilisateur a cliqué sur "voir la photo", il reçoit en retour dans son ordinateur la photo d'un acteur de théâtre interprétant le rôle de Julien Sorel. S'il a cliqué sur "Entendre la voix", il reçoit en retour un échantillon numérisé de la voix de l'acteur de théâtre interprétant un morceau de texte issu de la pièce.

On va maintenant décrire en relation avec la **FIG. 5** un système informatique (10) ou ordinateur apte à mettre en œuvre un procédé de traitement d'un document informatique selon l'invention. Ce système informatique est par exemple un micro-ordinateur ou une station de travail.

Dans ce mode de réalisation, le procédé de traitement d'un document selon l'invention est mis en œuvre par l'exécution d'un programme informatique selon l'invention chargé dans ce système informatique.

Le système informatique 10 comporte classiquement un bus de communication 112 auquel sont reliés :

- Une unité centrale 100 (microprocesseur).
- Une mémoire ROM 101, dans lequel peut être mémorisé le code exécutable du programme informatique de l'invention.
- Une mémoire RAM 102, comportant des registres adaptés à enregistrer des variables modifiées au cours de l'exécution du programme précité.
- Un écran 103 permettant de servir d'interface avec un utilisateur, à l'aide d'un clavier 104 ou de tout autre moyen, tel que par exemple une souris 105. Cet écran pourra permettre à un utilisateur de visualiser, par ex. au travers d'un logiciel de navigation sur Internet (navigateur Web), un document informatique tel qu'un document XML.
- Un disque dur 106, dans lequel on peut stocker, de façon alternative à la mémoire ROM, le code exécutable du programme selon l'invention. Le code source du programme pourra être également stocké dans le disque dur et compilé lors de l'exécution du programme.

- Un lecteur de disquette 107 adapté à recevoir une disquette 7. Le programme de l'invention stocké sur une disquette pourra ainsi alternativement être chargé dans l'ordinateur 10 via le lecteur de disquettes 107.

5 - Un lecteur de disques CD 108 adapté à lire un disque CD 8. Le programme de l'invention (source ou exécutable) stocké sur un disque CD pourra ainsi alternativement être chargé dans l'ordinateur 10 via le lecteur de disques CD 108.

10 - un lecteur de PC-CARD 109 pourra permettre de la même façon le chargement dans l'ordinateur du programme de l'invention stocké sur une carte PC-CARD 9.

15 - une interface de communication 110 avec un réseau de communication 1 permettant d'envoyer/recevoir des messages vers/depuis d'autre systèmes connectés au réseau. Il s'agit notamment, dans le cadre de l'invention, de messages de requête de documents (XML et XML Schema), de messages de requête d'exécution de fonctions distantes, et de messages de réponse suite à l'exécution de fonctions. La communication de messages sur le réseau utilise le protocole HTTP, selon le mode de réalisation préféré de l'invention. D'autre part, cette interface de communication 110 permet aussi de télécharger le programme de traitement de documents selon l'invention, depuis
20 un autre système informatique sur le réseau.

Il est à noter, d'une part, que le bus de communication 112 permet la communication entre les différents éléments inclus dans le système informatique 10 ou reliés à lui. D'autre part, la représentation du bus n'est pas limitative et, notamment, l'unité centrale (100) est susceptible de communiquer
25 des instructions à tout élément de l'ordinateur 10 directement ou par l'intermédiaire d'un autre élément de cet ordinateur.

En résumé, l'invention décrite ici concerne un procédé et un dispositif de traitement d'un document informatique dans un système informatique, ce dernier pouvant être relié à un réseau de communication. Selon le mode de réalisation préféré, les étapes du procédé sont déterminées par les instructions
30 d'un programme de traitement d'un message selon l'invention, et le procédé est

mis en œuvre lorsque ce programme est chargé dans un ordinateur dont le fonctionnement est alors commandé par l'exécution du programme.

En conséquence, l'invention s'applique également à un programme d'ordinateur, particulièrement un programme d'ordinateur sur ou dans un support d'informations, adapté à mettre en œuvre l'invention. Ce programme peut utiliser n'importe quel langage de programmation, et être sous la forme de code source, code objet, ou de code intermédiaire entre code source et code objet tel que dans une forme partiellement compilée, ou dans n'importe quelle autre forme souhaitable pour implémenter un procédé selon l'invention. Le support d'informations peut être n'importe quelle entité ou dispositif capable de stocker le programme.

Par exemple, le support peut comporter un moyen de stockage, tel qu'une ROM, par exemple un CD ROM ou une ROM semi-conducteur, ou un moyen d'enregistrement magnétique, par exemple une disquette (*floppy disc*) ou un disque dur. D'autre part, le support peut être un support transmissible tel qu'un signal électrique ou optique, qui peut être acheminé via un câble électrique ou optique, ou encore par radio ou par d'autres moyens.

Alternativement, le support peut être un circuit intégré dans lequel le programme est incorporé, le circuit étant adapté pour exécuter le procédé en question, ou pour être utilisé dans l'exécution du procédé.

Bien entendu, la présente invention ne se limite pas aux détails des formes de réalisation décrits ici à titre d'exemple, mais s'étend au contraire aux modifications à la portée de l'homme de l'art, sans sortir du cadre de l'invention.

REVENDICATIONS

1. Procédé de traitement d'une portion d'un document informatique
5 dans un système informatique, le contenu du document informatique étant
représenté avec un langage de balisage, chaque balise comportant un nom et
une valeur, ladite valeur pouvant inclure d'autres balises, le document
informatique étant associé à un second document informatique, appelé
10 "document schéma", dont le contenu est représenté dans un langage de
balisage de type schéma, ledit document schéma définissant la structure de
ladite portion du document informatique, le procédé de traitement étant
caractérisé en ce qu'il comporte les étapes suivantes :

- (A) – sélection (E13) d'une balise, dite "balise courante", dans ladite
portion du document informatique ;
- 15 (B) – recherche (E15-E21) dans le document schéma d'au moins une
balise déclarative d'une fonction associée à la balise sélectionnée du document
informatique ;
- (C) – création (E23, E25) d'une liste de fonctions applicables à ladite
balise courante dudit document informatique à partir de ladite au moins une
20 balise déclarative d'une fonction, trouvée dans ledit document schéma.

2. Procédé selon la revendication 1, dans lequel le contenu du
document informatique est affiché sur un écran d'affichage du système
informatique, la balise courante étant sélectionnée par un utilisateur sur l'écran
25 d'affichage au moyen d'un dispositif de commande, par exemple un dispositif
de pointage, le procédé comportant en outre l'étape suivante :

- (D) – affichage (E27) de la liste de fonctions applicables à la balise
courante dudit document informatique.

3. Procédé selon la revendication 1 ou 2, dans lequel la structure de chacune des balises contenues dans ladite portion du document informatique est définie dans le document schéma par des balises appropriées dans lesquelles un champ particulier contient le nom de la balise correspondante du document informatique.

4. Procédé selon la revendication 3, dans lequel l'étape de recherche (B) d'au moins une balise déclarative d'une fonction associée à la balise courante dans le document informatique, comprend les sous-étapes suivantes :

10 (B1) – recherche (E19) dans le document schéma d'une balise comportant le nom de la balise courante ;

(B2) – recherche (E21) de balises déclaratives de fonction, en liaison avec la balise trouvée à l'étape (B1).

15 5. Procédé selon l'une quelconque des revendications 1 à 4, dans lequel l'étape (C) de création d'une liste de fonctions applicables à ladite balise courante du document informatique, inclut les sous-étapes suivantes, effectuées pour chaque balise déclarative d'une fonction trouvée à l'étape (B):

(C1) – lecture (E231, E232) d'au moins une balise déclarative d'un argument de la fonction, ladite balise comportant un champ définissant le nom de l'argument, et un champ définissant le type de l'argument ;

20

(C2) – détermination (E232, E233) de la valeur de chaque argument par affectation audit argument de la valeur d'une balise du document informatique qui est identifiée selon un mode d'identification codé dans la balise déclarative de l'argument, dans le document schéma, ou bien par affectation audit argument d'une valeur fournie par l'utilisateur.

25

6. Procédé selon la revendication 5, dans lequel, à l'étape (C2), ledit mode d'identification d'une balise du document informatique, codé dans la balise déclarative de l'argument, dans le document schéma, consiste à

30

identifier dans le document informatique, la balise dont le nom est identique au nom défini dans la balise déclarative de l'argument dans le document schéma.

5 7. Procédé selon la revendication 5, dans lequel, à l'étape (C2), ledit mode d'identification d'une balise du document informatique, codé dans la balise déclarative de l'argument, dans le document schéma, consiste à fournir le chemin (Figs. 3a, 3b), dans le document informatique, à partir de la balise courante, jusqu'à la balise dont la valeur est à affecter audit argument.

10 8. Procédé selon l'une quelconque des revendications 2 à 7, comportant en outre l'étape suivante :

 (E) – exécution (E29) d'au moins une fonction de ladite liste affichée, en réponse à une commande générée par une action de l'utilisateur sur un dispositif de commande tel qu'un clavier d'ordinateur ou un dispositif de
15 pointage.

 9. Procédé selon la revendication 8, dans lequel ladite fonction est exécutée à distance (E291, E293, E294) sur un second système informatique relié audit système informatique au travers d'un réseau de communication.

20

 10. Procédé selon l'une quelconque des revendications précédentes, dans lequel chaque balise déclarative d'une fonction inclut en outre l'adresse électronique du code exécutable de la fonction, ou bien inclut directement le code exécutable de la fonction, ledit code exécutable pouvant être sous forme
25 de code source, code objet, ou de code intermédiaire entre code source et code objet.

 11. Procédé selon l'une quelconque des revendications précédentes, dans lequel, le contenu dudit document informatique est représenté en langage
30 XML, et ledit contenu dudit document schéma est représenté en langage XML Schema.

12. Procédé selon la revendication 11, dans lequel une balise déclarative de fonction est exprimée dans le document schéma de la façon suivante :

5 <xs:function name="..." text="..."/>

où :

"function" est le mot clé identifiant une balise de déclaration de fonction,

10 "name=" est un mot clé identifiant un champ dont la valeur définit le nom de la fonction,

"text=" est un mot clé identifiant un champ optionnel dont la valeur est du texte destiné à être affiché pour l'utilisateur, lors de l'affichage de la liste de fonctions.

15 13. Procédé selon la revendication 11 ou 12, dans lequel des balises déclaratives d'un argument sont exprimées dans le document schéma de la façon suivante :

 <xs:argument name="..." type="..."/>

...

20 <xs:argument name="..." type="..."/>

où :

"argument" est le mot clé identifiant une balise de déclaration d'un argument,

25 "name=" est un mot clé identifiant un champ dont la valeur définit le nom de l'argument,

"type=" est un mot clé identifiant un champ dont la valeur définit le type de l'argument.

30 14. Procédé selon la revendication 13, dans lequel une balise déclarative d'un argument comporte un champ supplémentaire identifié par un mot clé (*FromTag*), dont la valeur est représentative d'un chemin, dans ladite

portion du document informatique, permettant de trouver la balise dont la valeur est à affecter audit argument.

15. Procédé de traitement d'une portion d'un document informatique
5 dans un système informatique, le contenu du document informatique étant représenté avec un langage de balisage, chaque balise comportant un nom et une valeur, ladite valeur pouvant inclure d'autres balises, le document informatique étant associé à un second document informatique, appelé "document schéma", dont le contenu est représenté dans un langage de
10 balisage de type schéma, ledit document schéma définissant la structure de ladite portion du document informatique, le procédé de traitement étant caractérisé en ce qu'il comporte les étapes suivantes :

- (A) – sélection d'une balise, dite "balise courante", dans ladite portion du document informatique ;
- 15 (B) – recherche dans le document schéma d'une balise déclarative d'une fonction prédéterminée, associée à la balise sélectionnée du document informatique ;
- (C) – exécution de ladite fonction prédéterminée.

20 16. Procédé selon la revendication 15, dans lequel la structure de chacune des balises contenues dans ladite portion du document informatique est définie dans le document schéma par des balises appropriées dans lesquelles un champ particulier contient le nom de la balise correspondante du document informatique.

25 17. Procédé selon la revendication 16, dans lequel chaque balise déclarative de fonction contenue dans le document schéma comporte un champ prédéfini (*name*="...") dont la valeur définit le nom de la fonction, et dans lequel l'étape de recherche (B) d'une balise déclarative d'une fonction
30 prédéterminée associée à la balise courante dans le document informatique, comprend les sous-étapes suivantes :

(B1) – recherche dans le document schéma d'une balise comportant le nom de la balise courante ;

(B2) – recherche d'une balise déclarative de fonction, en liaison avec la balise trouvée à l'étape (B1), ladite balise déclarative de fonction contenant le
5 nom de ladite fonction prédéterminée.

18. Procédé selon la revendication 17, dans lequel l'étape (C) d'exécution de ladite fonction prédéterminée, inclut les sous-étapes suivantes :

10 (C1) – lecture d'au moins une balise déclarative d'un argument associée à ladite balise déclarative de fonction contenant le nom de ladite fonction prédéterminée, ladite balise comportant un champ définissant le nom de l'argument, et un champ définissant le type de l'argument ;

(C2) – détermination de la valeur de chaque argument par affectation
15 audit argument de la valeur d'une balise du document informatique qui est identifiée selon un mode d'identification codé dans la balise déclarative de l'argument, dans le document schéma.

19. Procédé selon l'une quelconque des revendications 15 à 18, dans
20 lequel ladite fonction prédéterminée est exécutée à distance sur un second système informatique relié audit système informatique au travers d'un réseau de communication.

20. Système informatique, relié ou non à un réseau de communication,
25 caractérisé en ce qu'il comporte des moyens adaptés à la mise en œuvre d'un procédé de traitement d'un document informatique, conforme à l'une quelconque des revendications précédentes.

21. Programme d'ordinateur sur un support d'informations, caractérisé en
30 ce qu'il comporte des instructions de programme adaptées à la mise en œuvre d'un procédé de traitement d'un document informatique selon l'une quelconque

des revendications 1 à 19, lorsque ce programme d'ordinateur est chargé et exécuté dans un système informatique.

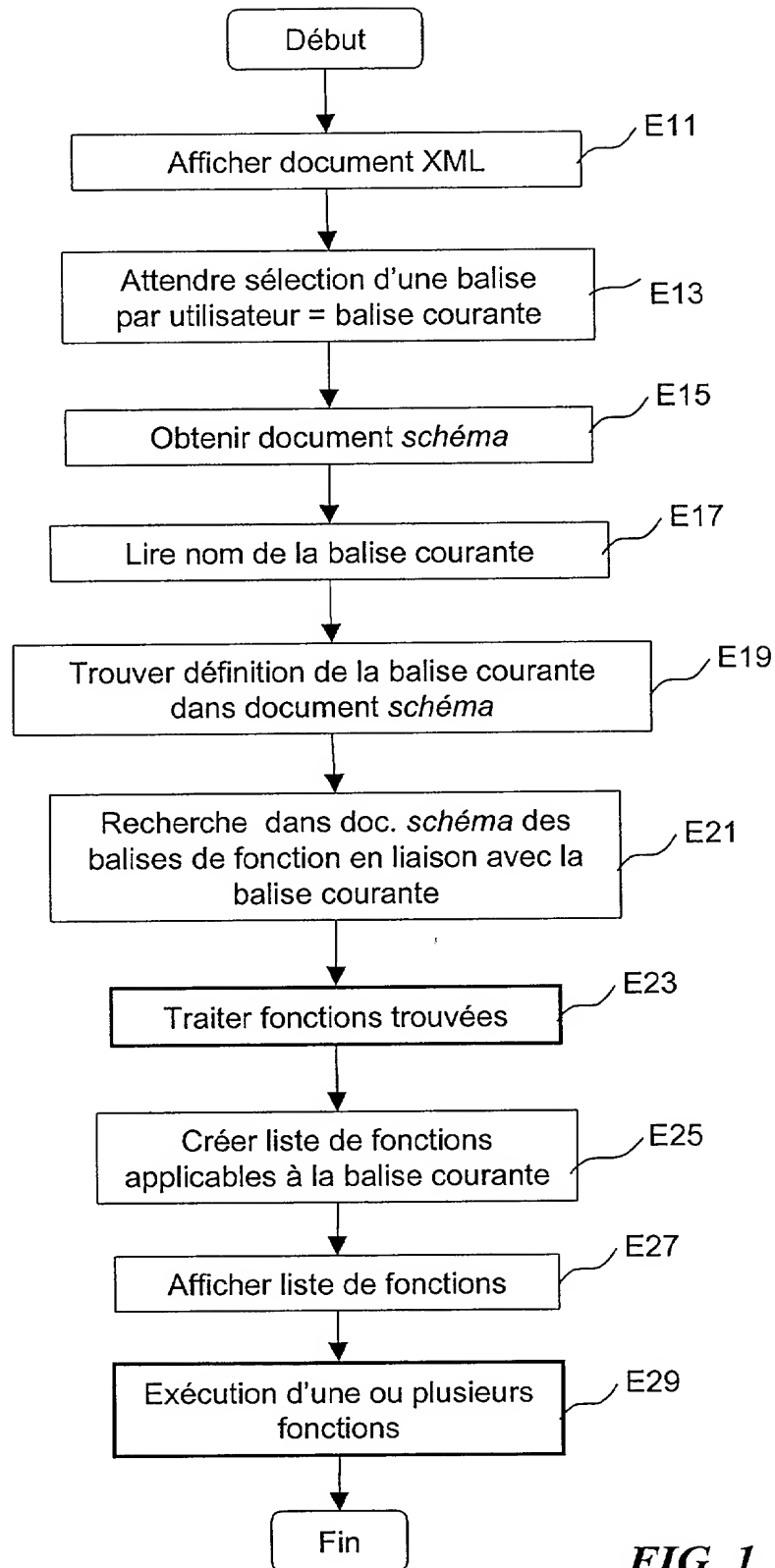


FIG. 1

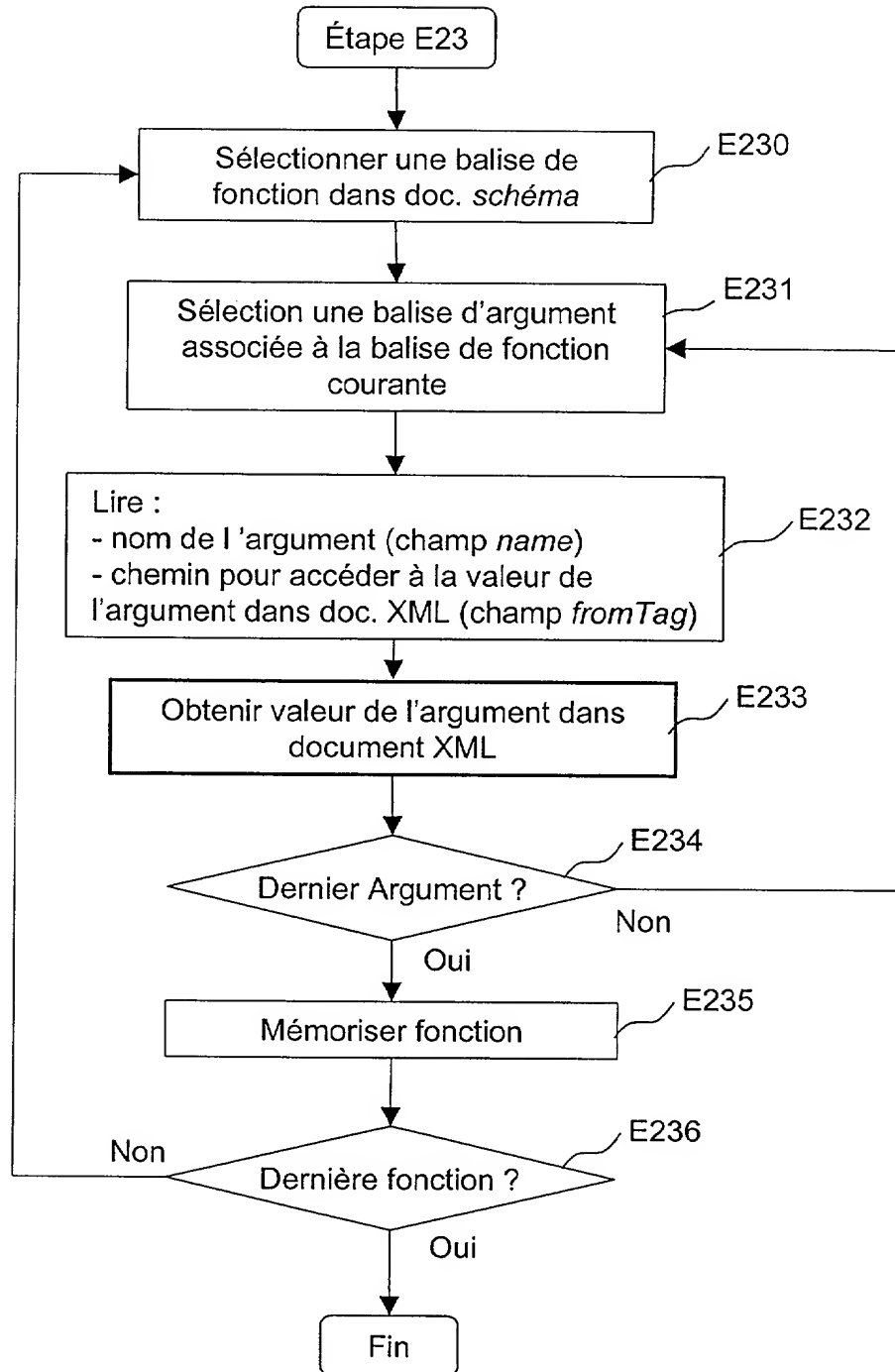


FIG. 2

3 / 5

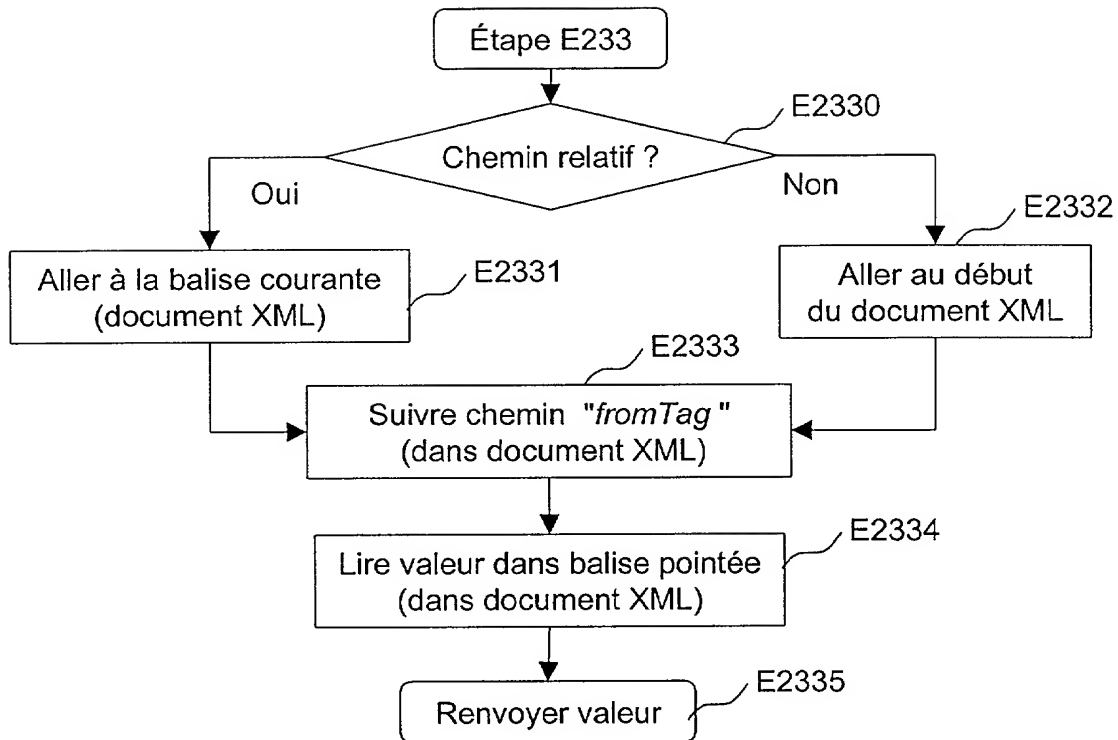


FIG. 3a

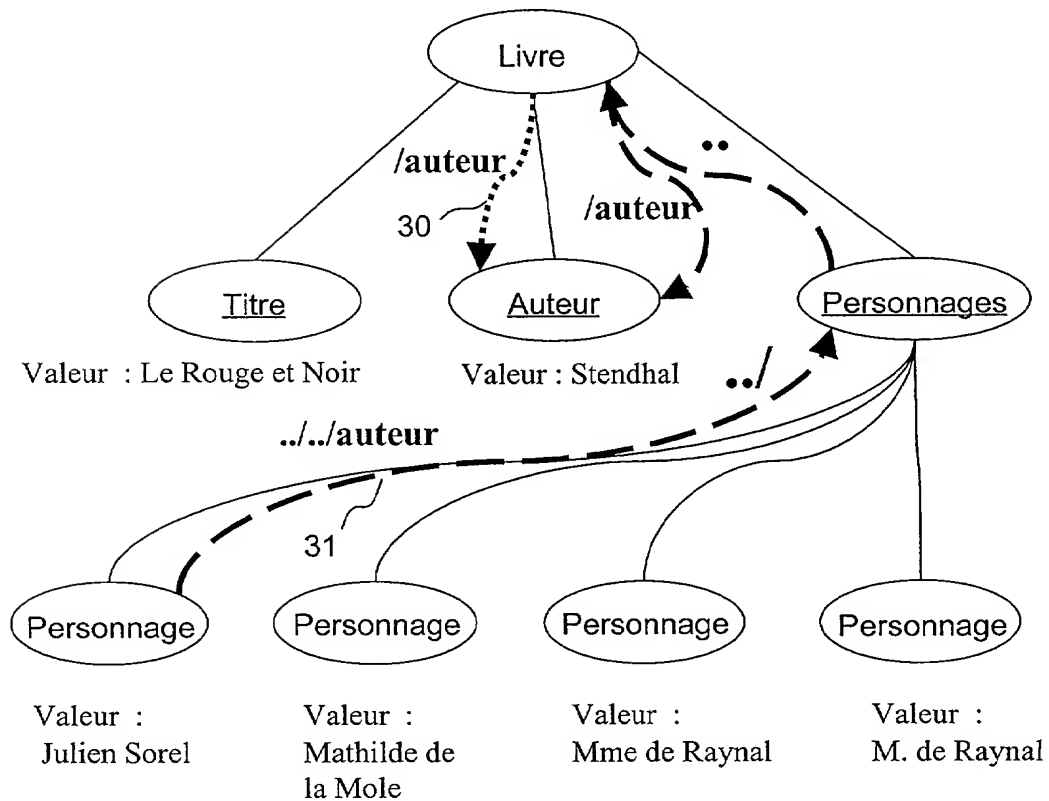
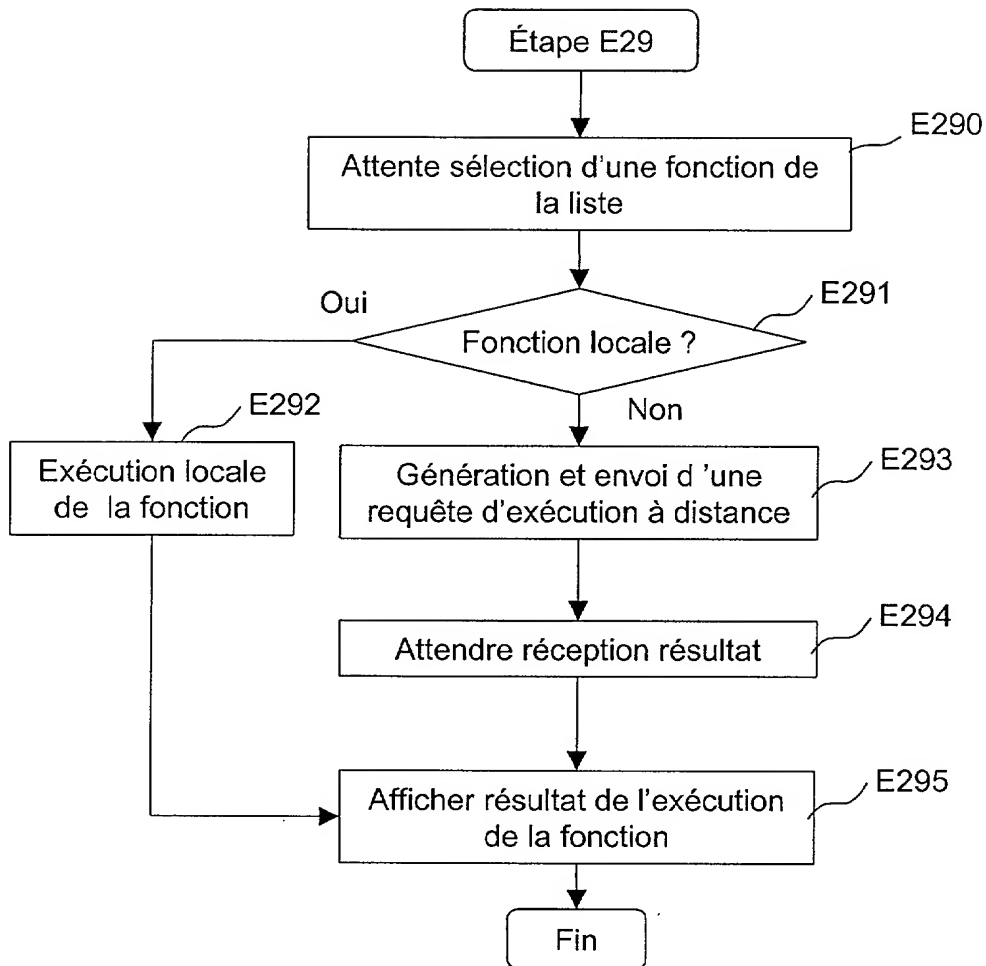


FIG. 3b

**FIG. 4**

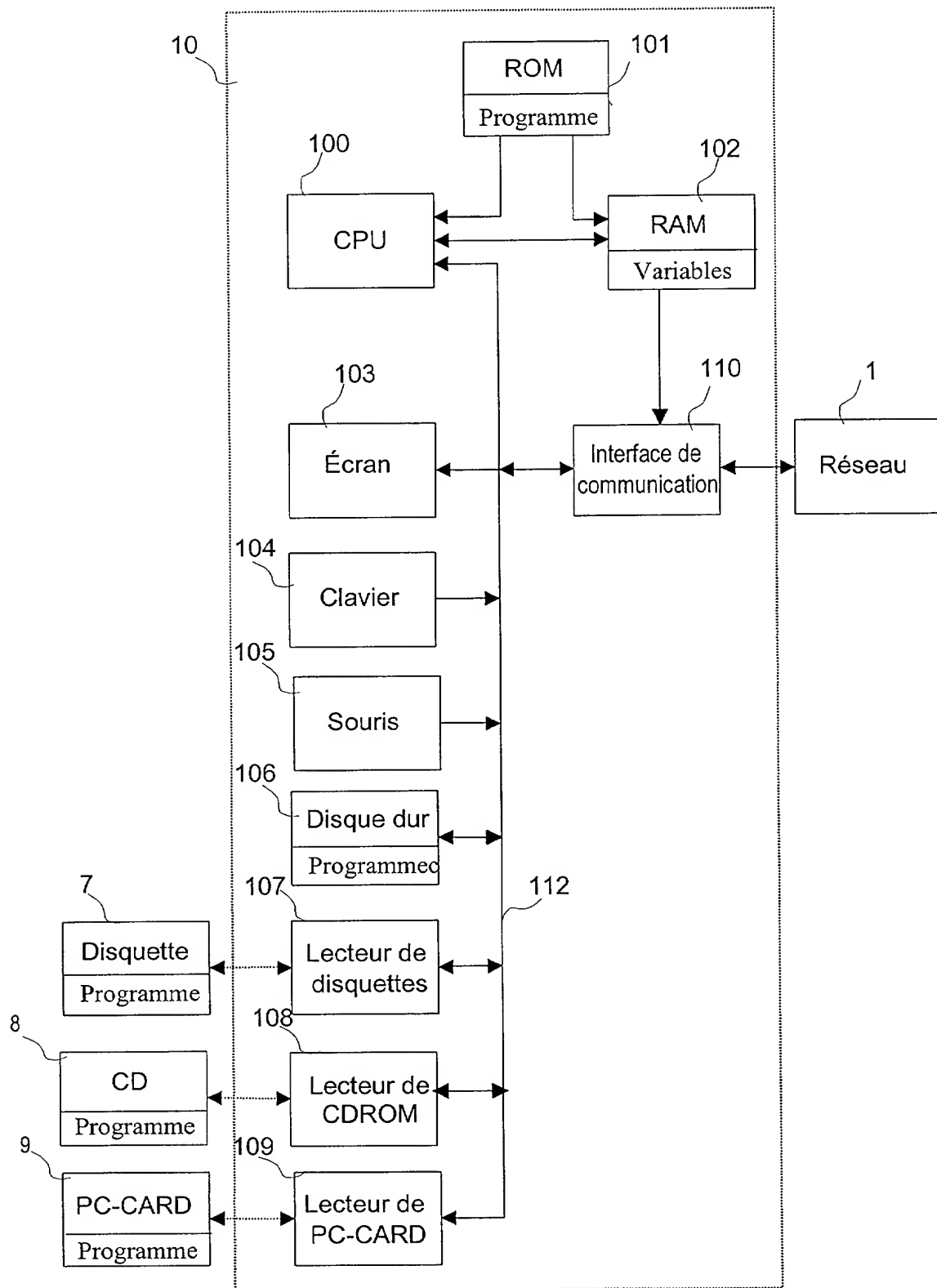


FIG. 5



RAPPORT DE RECHERCHE PRÉLIMINAIRE

établi sur la base des dernières revendications
déposées avant le commencement de la recherche

2826753

N° d'enregistrement
nationalFA 605932
FR 0108699

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
X	L. BOMPANI, P. CIANCARINI AND F. VITALI: "Software Engineering and the Internet: a roadmap" PROCEEDINGS OF THE CONFERENCE ON THE FUTURE OF SOFTWARE ENGINEERING, 2000, pages 305-315, XP002202432 Limerick, Ireland * page 306, colonne de gauche, ligne 3 - ligne 37; figure 2 * * page 308, colonne de gauche, ligne 9, alinéa 3 - page 311, ligne 15 *	1-6, 8-10, 15-21	
A	BATTLE S A ET AL: "Flexible information presentation with XML" IEE COLLOQUIUM MULTIMEDIA DATABASES AND MPEG-7, IEE, LONDON, GB, 29 janvier 1999 (1999-01-29), pages 13-1-13-6, XP002128574 * page 13-2, ligne 12 - page 13-6, ligne 7 *	1,3-6, 10, 15-18, 20,21	
A	MCKEE P ET AL: "Behavioural specification using XML" DISTRIBUTED COMPUTING SYSTEMS, 1999. PROCEEDINGS. 7TH IEEE WORKSHOP ON FUTURE TRENDS OF CAPE TOWN, SOUTH AFRICA 20-22 DEC. 1999, LOS ALAMITOS, CA, USA, IEEE COMPUT. SOC, US, 20 décembre 1999 (1999-12-20), pages 53-59, XP010367828 ISBN: 0-7695-0468-X * page 53, colonne de droite, alinéa 2 - page 54, colonne de gauche, alinéa 3 *	1,11	DOMAINES TECHNIQUES RECHERCHÉS (Int.CL.7) G06F
A	WO 00 68828 A (CLAUSSE JULIEN) 16 novembre 2000 (2000-11-16) * revendication 1; figure 1 *	1,15	
Date d'achèvement de la recherche		Examineur	
17 juin 2002		Fournier, C	
<p>CATÉGORIE DES DOCUMENTS CITÉS</p> <p>X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire</p> <p>T : théorie ou principe à la base de l'invention E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure. D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant</p>			

